

CLAIMS

1. (Currently Amended) A method comprising:

receiving a playlist at a media player referencing a first media segment and a second media segment, the second media segment operable to play automatically without a prompt after being loaded into an interactive media presentation control, the media player being configured to only present one media segment through a user interface at a time;

presenting the first media segment via the user interface; and

prerolling the second media segment, wherein prerolling the second media segment comprises:

loading at least a predetermined minimum portion of the second media segment into a buffer for the interactive media presentation control before the presenting of the first media segment is complete; and

immediately postponing presentation of the second media segment, the immediately postponing facilitates prevention of overlapping playback with the first media segment;

receiving an event from the interactive media presentation control indicating that the presenting of the first media segment is finished;

receiving a loading complete event from the interactive media presentation control indicating that the second media segment has been completely loaded; and

immediately presenting the second media segment via the user interface after receiving the event and the loading complete event, the immediately presenting

facilitating a seamless transition from the first media segment to the second media segment.

2. (Canceled).

3. (Currently Amended) A method as recited in claim 1 [[2]] wherein the postponing playback comprises:

issuing a stop command to a control.

4. (Currently Amended) A method as recited in claim 1 [[2]] wherein the postponing playback comprises:

stopping a timer associated with presenting the second media segment.

5. (Canceled).

6. (Currently Amended) A method as recited in claim 1 [[2]] further comprising:

~~receiving a loading complete event indicating that the second media segment has been completely loaded; and~~

in response to receiving the loading complete event, prerolling a third media segment.

7. (Original) A method as recited in claim 1 wherein the second media segment comprises an interactive media segment.

8. (Original) A method as recited in claim 1 wherein the second media segment is further operable to issue a custom event.

9. (Original) A method as recited in claim 8 wherein the custom event references a third media segment to be played in response to the custom event.

10. (Currently Amended) A method as recited in claim 1 [[5]] further comprising receiving an end of playback event from the second media segment.

11. (Currently Amended) A method comprising:

parsing a playlist of media segments at a host application comprising a media player having at least one reference to an interactive media segment operable to play continuously and a media presentation control operable to play the interactive media segment, the media player configured to only present one media segment at a time;

prerolling the interactive media segment in the media presentation control;
immediately stopping playback of the interactive media segment if it is not a first media segment in the playlist, the immediately stopping playback for preventing overlapping playback with a preceding media segment;

playing the interactive media segment in an interface of a the media player after receiving an event indicating that the presenting of the preceding media segment is complete and the prerolling is complete host application with a control operable to play the interactive media segment; and

receiving a media segment event from the media presentation control indicating that the playing of the interactive media segment has finished.

12. (Original) A method as recited in claim 11 further comprising:
stopping playback of the interactive media segment; and
playing a subsequent media segment referenced by a subsequent reference in the playlist.
13. (Currently Amended) A method as recited in claim 11 wherein the playlist comprises an Advanced Stream Redirector (ASX) ASX file.
14. (Original) A method as recited in claim 11 further comprising:
issuing to the host application a host-recognized event corresponding to the media segment event.
15. (Original) A method as recited in claim 14 wherein the media segment event comprises an EndOfPlayback event and the host-recognized event comprises a WMPEndOfPlayback event.
16. (Original) A method as recited in claim 11 wherein the media segment event comprises a custom event.
17. (Original) A method as recited in claim 11 further comprising:
playing a first media segment prior to the interactive media segment; and
buffering the interactive media segment in memory prior to completion of the first media segment.

18. (Original) A method as recited in claim 17 further comprising:

receiving a buffer progress indication from the control, the buffer progress indication indicating that a predetermined minimum portion of the interactive media segment has been buffered;

issuing an EndOfBuffering event to the host application.

19. (Original) A method as recited in claim 18 wherein the predetermined minimum portion is 100% of the interactive media segment.

20. (Original) A method as recited in claim 18 wherein the predetermined minimum portion is less than 100% of the interactive media segment.

21. (Original) A method as recited in claim 18 further comprising:

receiving a buffer complete indicator from the control indicating that 100% of the interactive media segment has been buffered;

issuing an EndOfStreaming event to the host application.

22. (Currently Amended) One or more [[A]] computer-readable storage media, storing medium comprising computer-executable processor-executable instructions that, when executed on a processor, to perform a method acts comprising:

instantiating an events wrapper associated with a ~~Flash~~® media segment created using vector-based graphics animation techniques;

initializing a control operable to playback the ~~Flash~~® media segment created using vector-based graphics animation techniques;

hosting the control in a portion of a user interface; ~~and~~

buffering the media segment created using vector-based graphics animation techniques prior to completion of playback of a previous media segment;

immediately postponing presentation of the media segment created using vector-based graphics animation techniques, the immediately postponing for preventing overlapping playback with the previous media segment; and

receiving notification from the control when the ~~Flash-®~~ media segment created using vector-based graphics animation techniques has completed playback.

23. (Currently Amended) The one or more [[A]] computer-readable storage media medium as recited in claim 22 wherein the method further comprises:

~~buffering the Flash-® media segment prior to completion of playback of a previous media segment;~~

receiving an end of buffering event from the control when the ~~Flash-®~~ media segment created using vector-based graphics animation techniques has finished buffering; and

playing the ~~Flash-®~~ media segment created using vector-based graphics animation techniques after when the previous media segment completes playback and the end of buffering event is received.

24. (Currently Amended) The one or more [[A]] computer-readable storage media medium of claim 22 wherein the method further comprises:

playing a later media segment after receiving the notification that the ~~Flash~~® media segment created using vector-based graphics animation techniques has completed playback.

25. (Currently Amended) A system comprising:

memory and a processor;

a media control module, stored in the memory and executable on the processor, operable to begin playing a media segment automatically after buffering the media segment; and

a host application module, stored in the memory and executable on the processor, operable to receive a reference to the media segment, initialize the media control module with the media segment, and cause the media control module to immediately postpone playing of the media segment after the media segment is buffered, the causing of the media control module to immediately postpone playing for preventing overlapping playback with an already playing media segment.

26. (Currently Amended) A system as recited in claim 25 further comprising:

an events wrapper module, stored in the memory and executable on the processor, operable to receive an end of buffering notification from the media control module and issue a corresponding end of buffering notification to the host application module.

27. (Currently Amended) A system as recited in claim 25 further comprising a playlist module, stored in the memory and executable on the processor, having one or more references to media segments to be played in sequential an order presented.

28. (Original) A system as recited in claim 27 wherein at least one of the referenced media segments comprises mixed media.

29. (Currently Amended) A system as recited in claim 27 wherein the playlist module comprises an event name associated with an event media segment to be played when a referenced media segment issues an event having the event name.

30. (Original) A system as recited in claim 25 wherein the playing of the media segment is postponed at least in part by stopping a timer that sends timer ticks to the media control for advancing playing of the media segment.